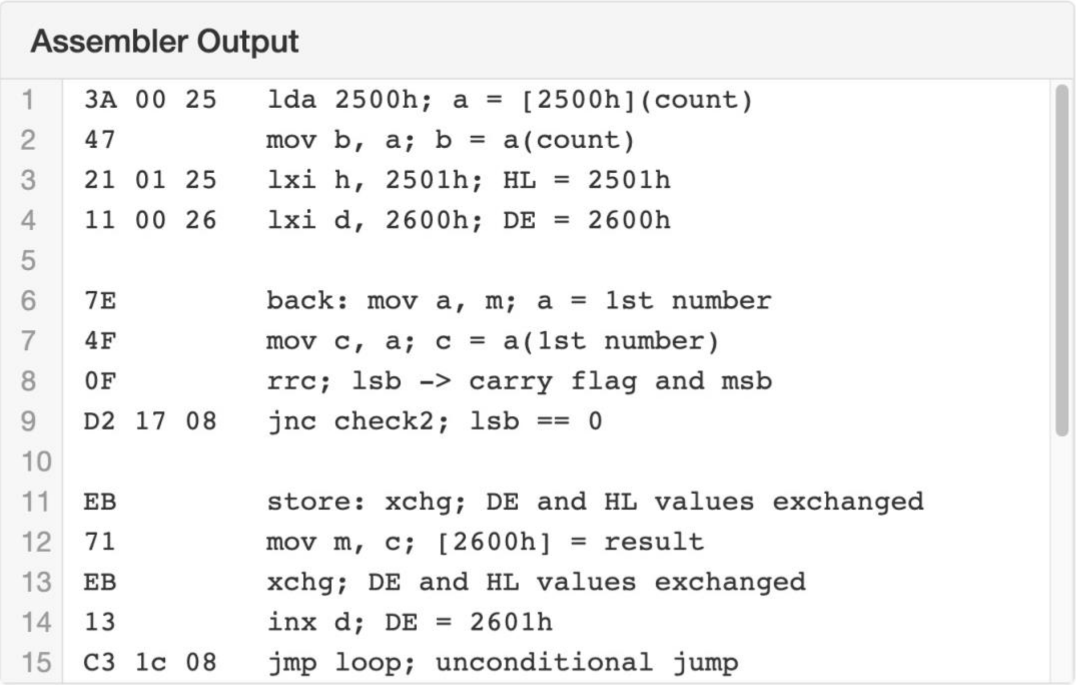
ASSIGNMENT 3

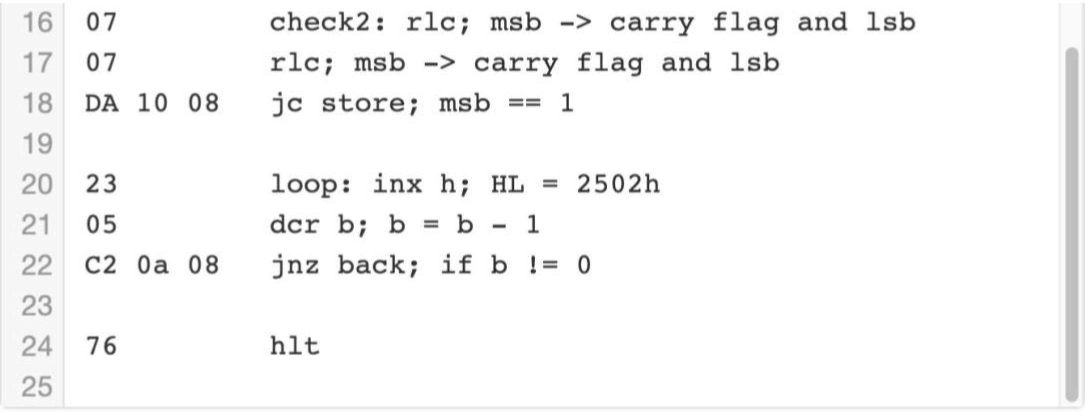
NAME: SAILIK PANDEY
ROLL: 002110501132
SEC-A3 BCSEII

1. A set of N data bytes is stored in m/m locations starting from 2501H. The value of N is stored in 2500H. Write a program to store these data bytes from m/m location 2600H if D0 or D7 is 1; otherwise reject the data byte.



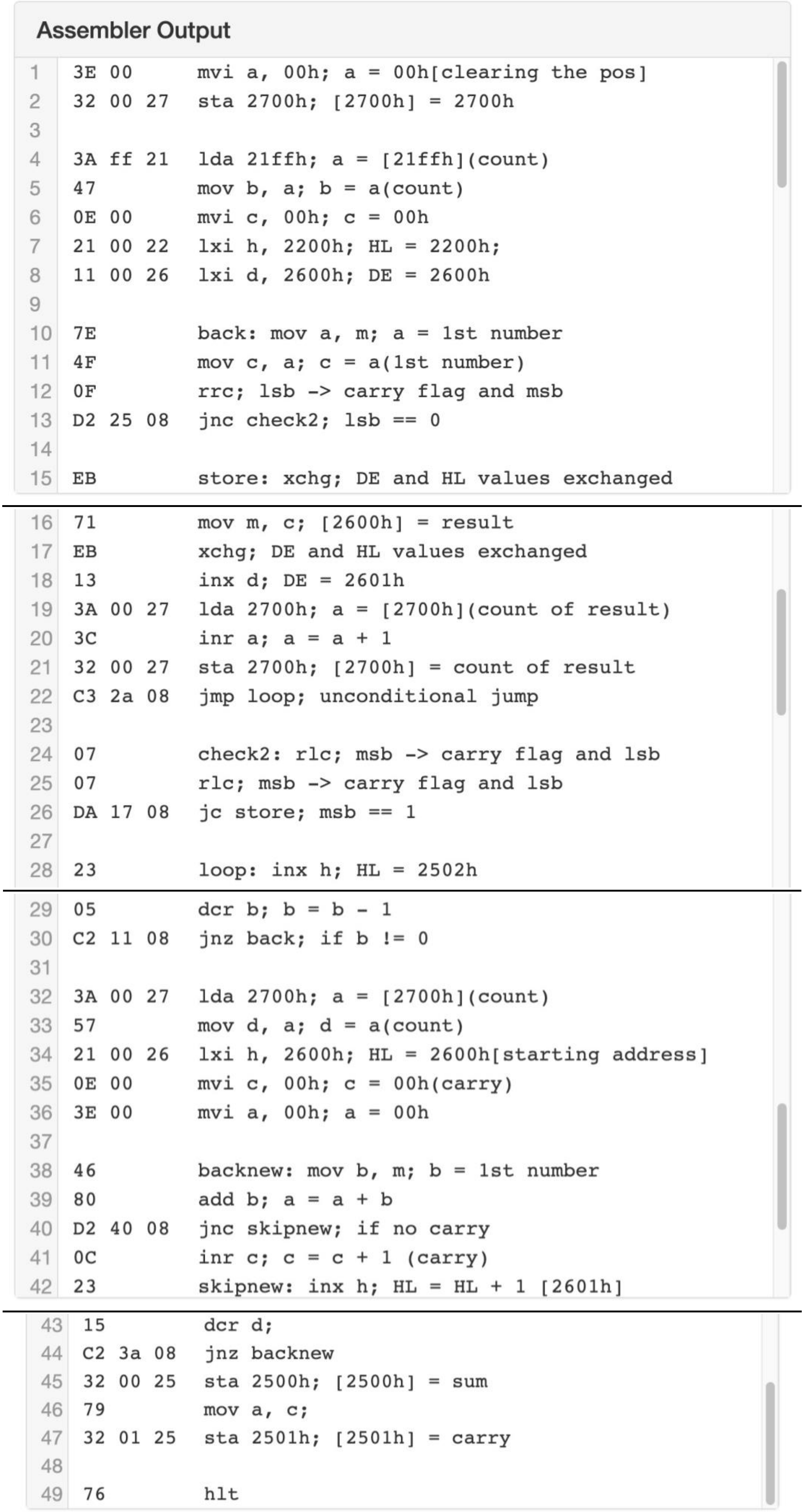
Assembler Output

```
3A 00 25   lda 2500h; a = [2500h](count)
47         mov b, a; b = a(count)
21 01 25   lxi h, 2501h; HL = 2501h
11 00 26   lxi d, 2600h; DE = 2600h

7E         back: mov a, m; a = 1st number
4F         mov c, a; c = a(1st number)
0F         rrc; lsb -> carry flag and msb
D2 17 08   jnc check2; lsb == 0

EB         store: xchg; DE and HL values exchanged
71         mov m, c; [2600h] = result
EB         xchg; DE and HL values exchanged
13         inx d; DE = 2601h
C3 1c 08   jmp loop; unconditional jump
```



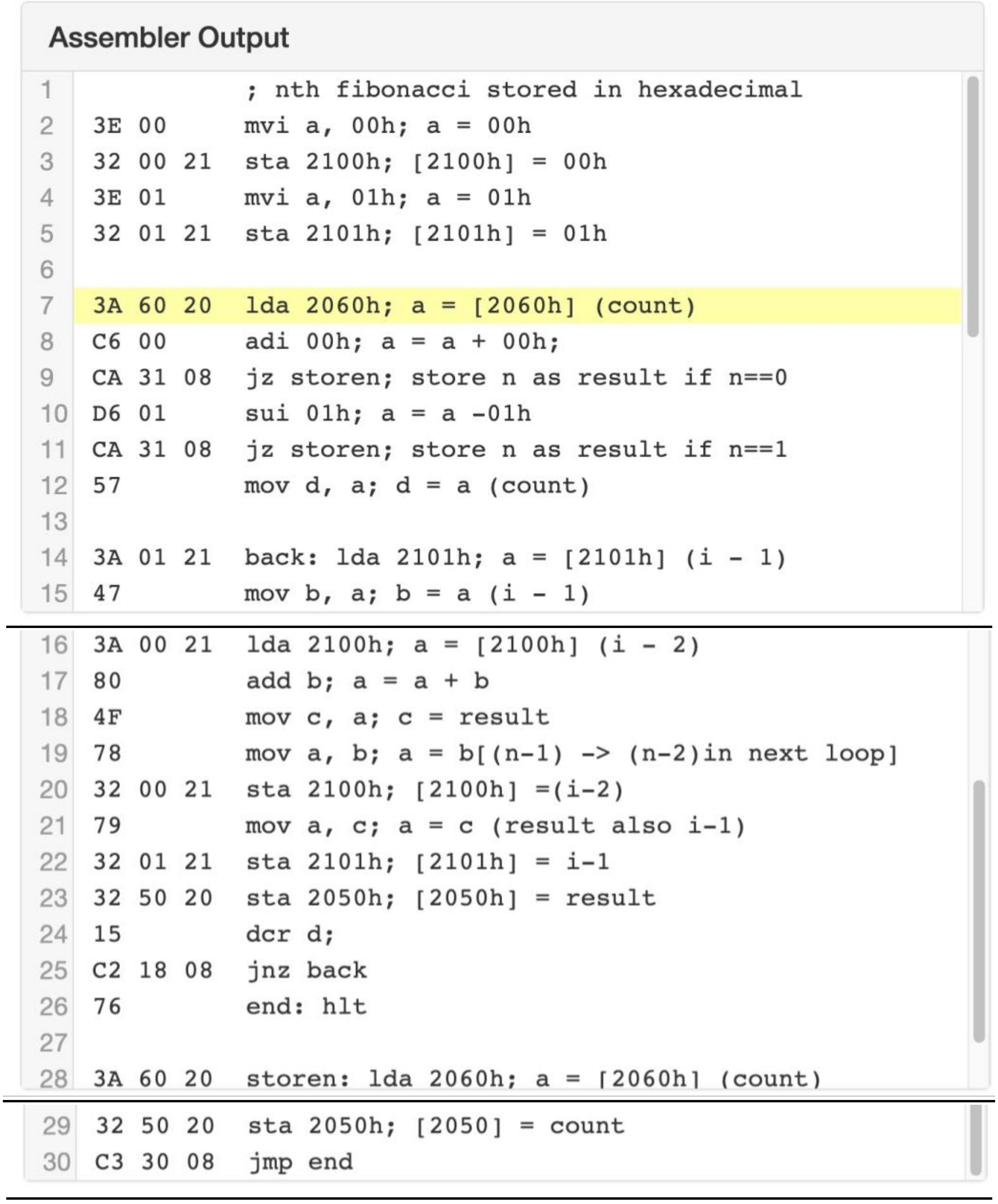
```
07         check2: rlc; msb -> carry flag and lsb
07         rlc; msb -> carry flag and lsb
DA 10 08   jc store; msb == 1

23         loop: inx h; HL = 2502h
05         dcr b; b = b - 1
C2 0a 08   jnz back; if b != 0

76         hlt

```

2. There are N data bytes stored from m/m location 2200H. The value of N is stored in 21FFH. Write an 8085 program to find the sum of integers whose LSB and MSB are 1. Store the result in 2500H and 2501H.



```
3E 00     mvi a, 00h; a = 00h[clearing the pos]
32 00 27  sta 2700h; [2700h] = 2700h

3A ff 21  lda 21ffh; a = [21ffh](count)
47        mov b, a; b = a(count)
0E 00     mvi c, 00h; c = 00h
21 00 22  lxi h, 2200h; HL = 2200h;
11 00 26  lxi d, 2600h; DE = 2600h

7E        back: mov a, m; a = 1st number
4F        mov c, a; c = a(1st number)
0F        rrc; lsb -> carry flag and msb
D2 25 08  jnc check2; lsb == 0

EB        store: xchg; DE and HL values exchanged
71        mov m, c; [2600h] = result
EB        xchg; DE and HL values exchanged
13        inx d; DE = 2601h
3A 00 27  lda 2700h; a = [2700h](count of result)
3C        inr a; a = a + 1
32 00 27  sta 2700h; [2700h] = count of result
C3 2a 08  jmp loop; unconditional jump

07        check2: rlc; msb -> carry flag and lsb
07        rlc; msb -> carry flag and lsb
DA 17 08  jc store; msb == 1

23        loop: inx h; HL = 2502h
05        dcr b; b = b - 1
C2 11 08  jnz back; if b != 0

3A 00 27  lda 2700h; a = [2700h](count)
57        mov d, a; d = a(count)
21 00 26  lxi h, 2600h; HL = 2600h[starting address]
0E 00     mvi c, 00h; c = 00h(carry)
3E 00     mvi a, 00h; a = 00h

46        backnew: mov b, m; b = 1st number
80        add b; a = a + b
D2 40 08  jnc skipnew; if no carry
0C        inr c; c = c + 1 (carry)
23        skipnew: inx h; HL = HL + 1 [2601h]
15        dcr d;
C2 3a 08  jnz backnew
32 00 25  sta 2500h; [2500h] = sum
79        mov a, c;
32 01 25  sta 2501h; [2501h] = carry

76        hlt
```

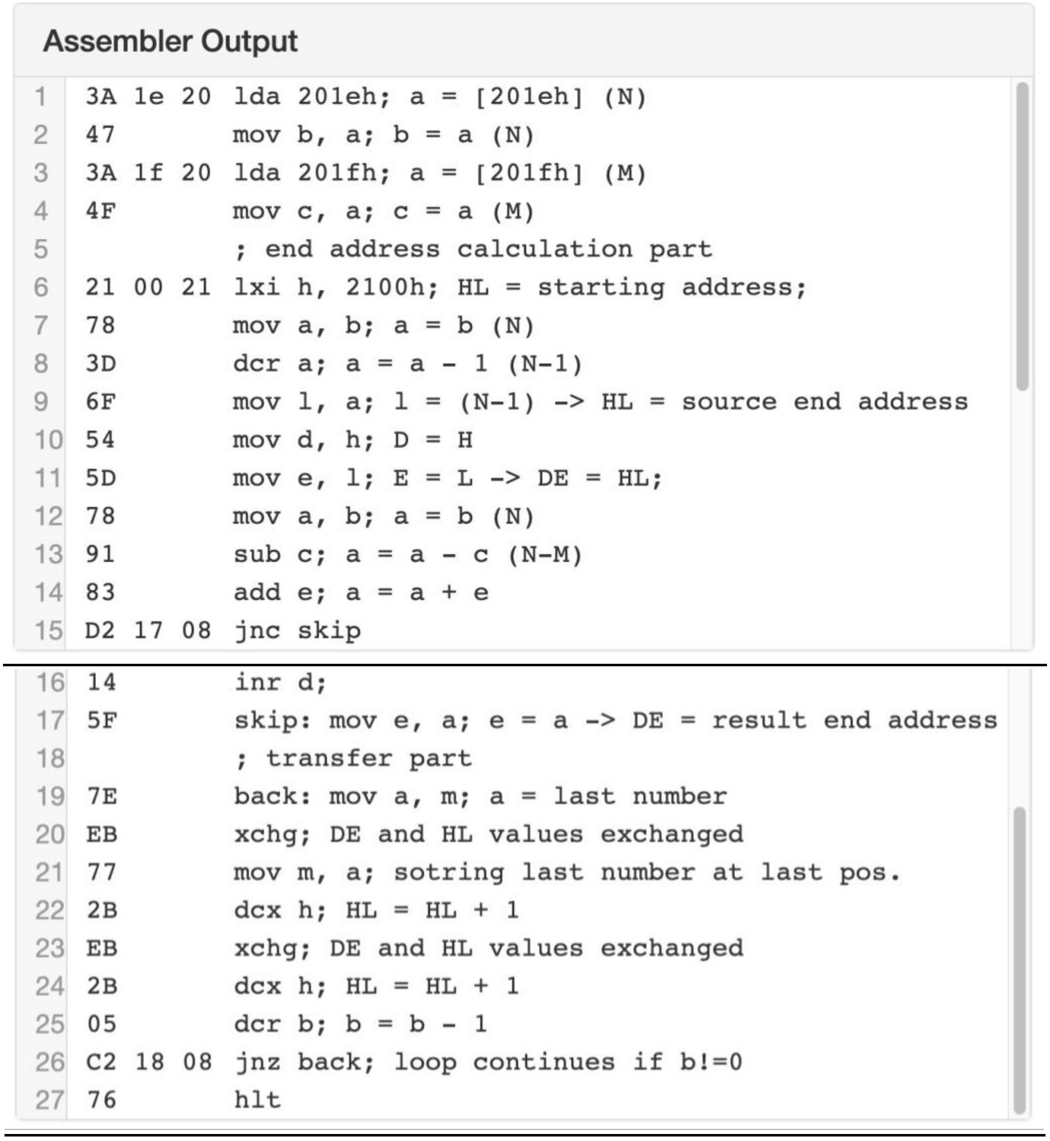
3. Write an 8085 program to generate Nth Fibonacci number using function and store it in 2050H. The value of N (8-bits) is stored in memory 2060H.



Assembler Output

```
                ; nth fibonacci stored in hexadecimal
3E 00           mvi a, 00h; a = 00h
32 00 21        sta 2100h; [2100h] = 00h
3E 01           mvi a, 01h; a = 01h
32 01 21        sta 2101h; [2101h] = 01h

3A 60 20        lda 2060h; a = [2060h] (count)
C6 00           adi 00h; a = a + 00h;
CA 31 08        jz storen; store n as result if n==0
D6 01           sui 01h; a = a -01h
CA 31 08        jz storen; store n as result if n==1
57              mov d, a; d = a (count)

3A 01 21        back: lda 2101h; a = [2101h] (i - 1)
47              mov b, a; b = a (i - 1)
3A 00 21        lda 2100h; a = [2100h] (i - 2)
80              add b; a = a + b
4F              mov c, a; c = result
78              mov a, b; a = b[(n-1) -> (n-2)in next loop]
32 00 21        sta 2100h; [2100h] =(i-2)
79              mov a, c; a = c (result also i-1)
32 01 21        sta 2101h; [2101h] = i-1
32 50 20        sta 2050h; [2050h] = result
15              dcr d;
C2 18 08        jnz back
76              end: hlt

3A 60 20        storen: lda 2060h; a = [2060h] (count)
32 50 20        sta 2050h; [2050] = count
C3 30 08        jmp end
```

4. Write a program to transfer a block of bytes of size N from location1 to location2 (location2 > location1) when the size of overlap between the two locations is defined by M. The values of N and M are stored in 201EH and 201FH, respectively.



```
3A 1e 20 lda 201eh; a = [201eh] (N)
47       mov b, a; b = a (N)
3A 1f 20 lda 201fh; a = [201fh] (M)
4F       mov c, a; c = a (M)
         ; end address calculation part
21 00 21 lxi h, 2100h; HL = starting address;
78       mov a, b; a = b (N)
3D       dcr a; a = a - 1 (N-1)
6F       mov l, a; l = (N-1) -> HL = source end address
54       mov d, h; D = H
5D       mov e, l; E = L -> DE = HL;
78       mov a, b; a = b (N)
91       sub c; a = a - c (N-M)
83       add e; a = a + e
D2 17 08 jnc skip
14       inr d;
5F       skip: mov e, a; e = a -> DE = result end address
         ; transfer part
7E       back: mov a, m; a = last number
EB       xchg; DE and HL values exchanged
77       mov m, a; sotring last number at last pos.
2B       dcx h; HL = HL + 1
EB       xchg; DE and HL values exchanged
2B       dcx h; HL = HL + 1
05       dcr b; b = b - 1
C2 18 08 jnz back; loop continues if b!=0
76       hlt
```